

Qlib : AI 기반 양적 투자 플랫폼

Xiao Yang , Weiqing Liu , Dong Zhou , Jiang Bian 및 Tie-Yan Liu

Microsoft Research {Xiao.Yang, Weiqing.Liu,

Zhou.Dong, Jiang.Bian, Tie-Yan.Liu}@microsoft.com

추상적인

양적 투자는 일련의 금융 상품에 대해 순차적인 거래 기간 동안 수익을 극대화하고 위험을 최소화하는 것을 목표로 합니다. 최근 퀀트 투자에서 놀라운 혁신을 창출하는 AI 기술의 빠른 발전과 큰 잠재력에 영감을 받아 퀀트 연구 및 실제 투자를 위한 AI 기반 워크플로우의 채택이 증가하고 있습니다. 정량적 투자 방법론을 풍부하게 하는 동안 AI 기술은 정량적 투자 시스템에 새로운 도전을 제기했습니다. 특히 양적 투자를 위한 새로운 학습 패러다임은 개선된 워크플로를 수용하기 위한 인프라 업그레이드를 요구합니다. 또한 AI 기술의 데이터 중심 특성은 실제로 더 강력한 성능을 갖춘 인프라가 필요함을 나타냅니다. 또한 재무 시나리오에서 다양한 작업을 해결하기 위해 AI 기술을 적용하는 데 몇 가지 고유한 문제가 있습니다. 이러한 문제를 해결하고 AI 기술과 양적 투자 사이의 격차를 해소하기 위해 우리는 잠재력을 실현하고, 연구를 강화하고, 양적 투자에서 AI 기술의 가치를 창출하는 것을 목표로 하는 Qlib를 설계 및 개발합니다.

관점. 첫째, AI 기술의 유연성으로 인한 양적 투자 워크플로의 기술 혁명은 새로운 지원 인프라를 요구하는 경향이 있습니다.

예를 들어 기존의 양적 투자는 일반적으로 전체 워크플로를 주시 추세 예측, 포트폴리오 최적화 등 몇 가지 하위 작업으로 분할하는 반면 AI 기술을 통해 최종 결과를 생성하는 엔드 투 엔드 솔루션을 구축할 수 있습니다. 포트폴리오를 직접 이러한 end-to-end 솔루션을 지원하기 위해서는 데이터 기반 특성으로 인해 현재 인프라를 업그레이드해야 합니다.

한편, AI 기술은 금융 분야의 풍부한 도메인 지식과 데이터 과학 분야의 풍부한 경험이 모두 필요한 몇 가지 새로운 시나리오에서 고유한 문제를 처리해야 합니다. 도메인 적응 없이 양적 연구 작업에 솔루션을 적용하는 것은 거의 작동하지 않습니다. 이러한 상황은 AI 시대에 이러한 현대적인 정량적 연구 워크플로우를 수용하고 금융 시나리오에서 AI 기술을 적용하기 위한 지침을 제공할 플랫폼에 대한 긴급한 요구로 이어집니다.

따라서 우리는 Qlib이라는 새로운 AI 지향 양적 투자 플랫폼을 제안합니다. 퀀트 투자에서 AI 기술의 큰 잠재력을 탐구하는 연구 노력을 지원하고 퀀트 연구원이 AI 기반 퀀트 투자에서 더 중요한 가치를 창출할 수 있도록 지원하는 것을 목표로 합니다. 특히 Qlib의 AI 지향 프레임워크는 AI 기반 솔루션을 수용하도록 설계되었습니다. 또한 퀀트 투자 시나리오 전용 고성능 인프라를 제공하여 많은 AI 연구 주제를 가능하게 합니다. 또한 양적 투자 시나리오에서 기계 학습을 위해 설계된 일련의 도구가 Qlib와 통합되어 사용자가 AI 기술을 최대한 활용할 수 있도록 합니다.

1. 소개

가장 뜨거운 연구 분야 중 하나인 양적 투자는 학계와 금융계에서 수많은 명석한 인재들을 끌어들이고 있습니다. 지난 수십 년 동안 퀀트 방법론을 최적화하기 위한 지속적인 노력으로 전체 전문 투자자 커뮤니티는 잘 정립되었지만 불완전한 퀀트 리서치 워크플로우를 요약했습니다. 최근 떠오르는 AI 기술은 이 연구 분야에서 새로운 트렌드를 시작합니다. 퀀트 투자에서 AI의 큰 잠재력을 탐구하는 데 대한 관심이 높아지면서 AI 기술은 퀀트 연구자들이 실용 분야에 널리 채택했습니다.

AI 기술은 정량적 투자 방법론을 풍부하게 하는 동시에 여러 측면에서 정량적 투자 시스템에 대한 새로운 도전을 제시합니다.

마지막으로 양적 투자의 일반적인 작업에 대한 여러 솔루션을 비교하여 몇 가지 사용 사례를 시연하고 Qlib 인프라의 성능을 평가합니다.

결과는 양적 투자 전용 Qlib의 인프라가 이 작업에서 대부분의 기존 솔루션을 능가함을 보여줍니다.

2 배경 및 관련 연구

이 섹션에서는 먼저 현대 양적 연구자가 응용할 때의 주요 실제 문제를 설명합니다.

1코드는 <https://github.com/microsoft/qlib>에서 확인할 수 있습니다.

Qlib 탄생의 원동력이 되는 양적 투자에 AI 기술을 접목한 그림 관련 업무를 간략하게 소개하겠습니다.

2.1 현실적인 문제

정량적 리서치 워크플로 혁명 전통적인 투자 리서치 워크플

로에서 연구원들은 종종 선형 모델(Petkova, 2006) 또는 수동으로 설계된 규칙(Murphy, 1999)에 의해 거래 신호를 개발합니다. 데이터. 그런 다음 대상 포트폴리오를 생성하기 위해 거래 전략(일반적으로 Barra[Sheikh, 1996])을 따릅니다. 마지막으로 연구원들은 백 테스트 기능을 통해 거래 신호와 포트폴리오를 평가합니다.

AI 기술의 부상과 함께 전통적인 양적 투자의 기술 혁명을 시작합니다. 전통적인 양적 연구 워크플로는 이러한 유연한 기술을 수용하기에는 너무 원시적입니다. 차이점을 보다 직관적으로 보여주기 위해 AI 기술을 기반으로 하는 전형적인 현대 연구 워크플로우를 시연합니다. 많은 기능(일반적으로 수백 차원 이상)이 있는 데이터 세트로 시작합니다. 이러한 양의 기능을 수동으로 설계하려면 많은 시간이 걸립니다. 기계 학습 알고리즘을 활용하여 이러한 기능을 자동으로 생성하는 것이 일반적입니다[Potvin et al., 2004; 널리 등, 1997; 알렌과 카잘라넨, 1999; Kakushadze, 2016]. 데이터 생성(Feng et al., 2019)은 데이터 세트를 구성하는 또 다른 옵션입니다. 연구원들은 다양한 데이터 세트를 기반으로 거래 신호를 마이닝하기 위해 수백 가지 기계 학습 방법을 제공했습니다[Sezer et al., 2019]. 연구원은 이러한 거래 신호를 기반으로 대상 포트폴리오를 생성할 수 있습니다. 그러나 그러한 워크플로만이 유일한 선택은 아닙니다. RL(reinforcement learning)은 작업을 여러 단계로 나누는 대신 데이터에서 최종 거래 행동까지 직접 end-to-end 솔루션을 제공합니다[Deng et al., 2016]. RL은 재무 시나리오의 거래 시뮬레이터인 환경과 상호 작용하여 거래 전략을 최적화합니다. RL은 기존 연구 워크플로의 백테스팅 기능 대신 반응형 시뮬레이터가 필요합니다.

또한 대부분의 AI 알고리즘에는 복잡한 하이퍼 매개변수가 있으므로 주의 깊게 조정해야 합니다.

AI 기술은 매우 유연하며 이미 전통적인 방법론을 위해 설계된 기존 도구의 범위를 넘어섭니다. 처음부터 AI 기술을 기반으로 연구 워크플로우를 구축하려면 많은 시간이 걸립니다.

인프라에 대한 고성능 요구 사항 AI 기술이 등장하면서 인프라에 대

한 요구 사항이 변경되었습니다. 이러한 데이터 기반 방법은 엄청난 양의 데이터를 활용할 수 있습니다. 초단타 거래 시나리오에서 데이터의 양은 TB 규모에 도달할 수 있습니다. 게다가 총 5개의 차원으로 구성된 기본 가격 및 볼륨 데이터에서 수천 개의 새로운 기능(예: Alpha101 [Kakushadze, 2016])을 도출하는 것이 매우 일반적입니다. 일부 연구자들은 표현을 검색하여 새로운 요소나 특징을 생성하려고 시도하기도 합니다[Allen and Karjalainen, 1999; 널리 등, 1997; 포트빈 등, 2004]. 데이터 처리의 이러한 과중한 작업은 연구원에게 과도한 부담을 주고 일부 연구를 수행합니다.

불가능한 주제. 이러한 상황에서는 인프라에 대한 보다 엄격한 성능 요구 사항이 제시됩니다.

머신러닝 솔루션 적용의 걸림돌 재무 데이터와 작업에는 고유성

과 과제가 있습니다. 적응 없이 기계 학습 솔루션을 정량적 연구 작업에 적용하는 것은 거의 효과가 없습니다. 금융 데이터의 SNR(Signal to Noise Ratio)이 매우 낮기 때문에 금융 시장에서 성공적인 데이터 기반 전략을 구축하기가 매우 어렵습니다. 대부분의 기계 학습 알고리즘은 데이터 기반이며 이러한 어려움을 처리해야 합니다. 없이

세부 사항을 주의 깊게 처리하면 기계 학습 모델이 만족스러운 성능을 거의 달성할 수 없습니다. 사소한 실수라도 효과적인 패턴을 학습하는 대신 모델이 노이즈에 과적합될 수 있습니다. 세부 사항을 올바르게 처리하려면 금융 산업에 대한 많은 도메인 지식이 필요합니다. 또한 연간 수익률과 같은 일반적인 목표는 구분할 수 없는 경우가 많기 때문에 기계 학습 방법에 대해 모델을 직접 교육하기가 어렵습니다. 적절한 감독 대상으로 합리적인 작업을 정의하는 것은 재무 데이터를 모델링하는 데 매우 중요합니다. 이러한 장벽은 금융 산업에 대한 도메인 지식이 많지 않은 많은 데이터 과학자를 위압합니다.

기계 학습 애플리케이션을 구축하는 데 필요한 또 다른 단계는 하이퍼파라미터 최적화입니다. 서로 다른 기계 학습 알고리즘에는 서로 다른 하이퍼파라미터 검색 공간이 있으며, 각 검색 공간에는 의미와 우선 순위가 다른 여러 차원이 있습니다. 일부 정량적 연구원은 전통적인 금융 산업에서 왔으며 기계 학습에 대한 지식이 많지 않습니다. 이러한 막대한 학습 비용으로 인해 많은 사용자가 기계 학습의 최대 가치를 충분히 발휘하지 못합니다.

2.2 관련 업무 금융 산업에

서 투자 전략은 투자자가 많을수록 수익성이 떨어집니다. 따라서 금융 종사자, 특히 양적 연구원은 자신의 알고리즘과 도구를 공유하는 데 열성적이지 않습니다. OLPS[Li et al., 2016]는 포트폴리오 선택을 위한 최초의 오픈 소스 도구 상자입니다. 새로운 학습 방법의 개발을 촉진하기 위한 벤치마크 및 툴킷으로서 기계 학습 알고리즘으로 구동되는 고전적인 전략 제품군으로 구성됩니다. 이 도구 상자는 Matlab 및 Octave만 지원하며 이는 현재 과학 주류 언어인 Python과 호환되지 않으므로 최신 기계 학습 알고리즘에 적합하지 않습니다. 그 프레임워크는 매우 간단하며 AI 기술을 기반으로 하는 현대의 양적 연구 워크플로우는 훨씬 더 복잡합니다. 최근 몇 년 동안 다른 정량적 도구가 등장했습니다. QuantLib[Firth, 2004]는 현대 정량 연구 워크플로우의 일부에만 초점을 맞춥니다. QUANTAXIS는 연구 워크플로우 대신 IT 인프라에 더 집중합니다. Quantopian은 일련의 오픈 소스 도구를 출시합니다. 1) Alphalens: 예측(알파) 주식 요인의 성능 분석을 위한 Python 라이브러리 2) Zipline: 백테스팅을 위한 이벤트 기반 시스템 3) Pyfolio: 성능 및 재무 포트폴리오의 위험 분석. 그들 모두는 거래 신호 또는 투자 포트폴리오 분석에만 중점을 둡니다.

2 시

²<https://github.com/QUANTAXIS/QUANTAXIS>

전반적으로 Qlib는 AI 시대의 현대 양적 연구자의 워크플로를 수용하는 최초의 오픈 소스 플랫폼입니다. 모든 퀀트 연구원이 퀀트 투자에서 AI 기술의 큰 잠재력을 실현할 수 있도록 지원하는 것을 목표로 합니다.

3 AI 중심의 정량적 투자 플랫폼

3.1 전반적인 디자인 금융 시

장에서 수년간 실무 경험을 쌓은 퀀트 연구원과 협력하여 위의 모든 문제를 해결하고 모든 종류의 솔루션을 탐색했습니다. 현재 상황에 동기를 부여하여 양적 투자에 AI 기술을 적용하기 위해 Qlib를 구현합니다.

AI 지향 프레임워크 Qlib는 최신 연구 워크플로를 기반으로 모듈화된 방식으로 설계되어 AI 기술을 수용할 수 있는 최대의 유연성을 제공합니다. 정량적 연구자는 모듈을 확장하고 아이디어를 효율적으로 시도하기 위한 워크플로를 구축할 수 있습니다. 각 모듈에서 Qlib는 실제 투자에서 매우 잘 작동하는 몇 가지 기본 구현 선택 사항을 제공합니다. 이러한 기성 모듈을 사용하면 정량적 연구자는 다른 사소한 세부 사항에 주의를 분산시키지 않고 특정 모듈에 관심이 있는 문제에 집중할 수 있습니다. 코드 외에도 계산 및 데이터도 일부 모듈에서 공유할 수 있으므로 Qlib는 사용자에게 도구 상자가 아닌 플랫폼으로 제공되도록 설계되었습니다.

고성능 인프라 AI 기술과 같은 데이터 기반 방법에는 데이터 처리 성능이 중요합니다. AI 지향 플랫폼인 Qlib는 고성능 데이터 인프라를 제공합니다. Qlib는 시계열 플랫폼 파일 데이터베이스를 제공합니다. 이러한 데이터베이스는 재무 데이터에 대한 과학적 컴퓨팅 전용입니다. 퀀트 투자 연구의 일부 일반적인 데이터 처리 작업에서 이 데이터베이스는 기존 데이터베이스 솔루션보다 성능이 훨씬 뛰어납니다. 또한 데이터베이스는 표현 계산에 의존하는 연구 주제를 가능하게 하는 요인/특징의 구현 및 계산을 가속화할 수 있는 표현 엔진을 제공합니다.

기계 학습에 대한 지침 Qlib는 일반적인 기계 학습 알고리즘이 일반화 능력으로 패턴을 완전히 학습할 수 있는 정량적 투자를 위한 몇 가지 일반적인 데이터 세트와 통합되었습니다. Qlib는 기계 학습 사용자를 위한 몇 가지 기본 지침을 제공하고 합리적인 기능 공간과 대상 레이블로 구성된 몇 가지 합리적인 작업을 통합합니다. 몇 가지 일반적인 하이퍼파라미터 최적화 도구가 제공됩니다. 지침과 합리적인 설정을 통해 머신 러닝 모델은 노이즈를 과적합하는 대신 더 나은 일반화 능력으로 패턴을 학습할 수 있습니다.

3.2 AI 지향 프레임워크

그림 1은 Qlib의 전체 프레임워크를 보여줍니다. 이 프레임워크는 1) 최신 AI 기술을 수용하고, 2)

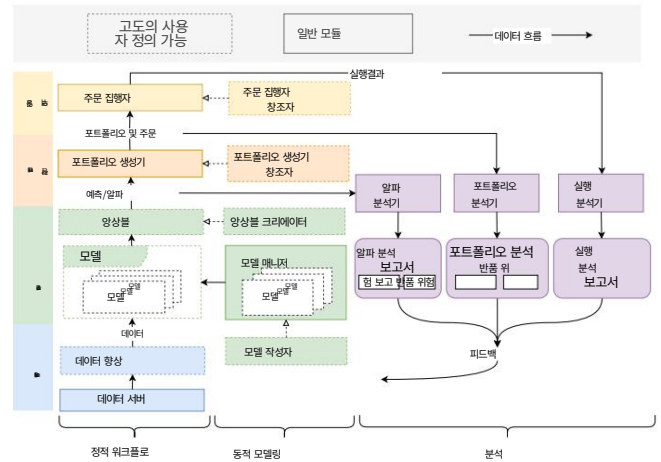


그림 1: Qlib로 구축된 모듈 및 일반적인 워크플로

퀀트 연구원이 최소한의 노력으로 전체 연구 작업 흐름을 구축하도록 돕고 3) 다른 부분에 방해받지 않고 관심 있는 문제를 탐색할 수 있는 최대한의 유연성을 제공합니다.

이러한 목표는 시스템 설계 관점에서 모듈화된 설계로 이어진다. 이 시스템은 최신 실용 연구 워크플로를 기반으로 여러 개별 모듈로 나뉩니다. 대부분의 퀀트 투자 리서치 방향은 전통적이든 AI 기반이든 관계없이 하나 또는 여러 모듈의 인터페이스 구현으로 간주될 수 있습니다. Qlib는 각 모듈의 사용자를 위한 실질적인 투자에서 잘 작동하는 몇 가지 일반적인 구현을 제공합니다. 또한 모듈은 연구원이 기존 방법을 재정의하여 새로운 아이디어를 탐색할 수 있는 유연성을 제공합니다. 이러한 프레임워크를 사용하여 연구원은 최소한의 비용으로 새로운 아이디어를 시도하고, 실험을 테스트할 수 있습니다.

Qlib의 모듈은 그림 1에 나열되어 있으며 일반적인 워크플로에 연결되어 있습니다. 각 모듈은 양적 투자의 일반적인 하위 작업에 해당합니다. 모듈의 구현은 이 작업에 대한 솔루션으로 간주될 수 있습니다. 각 모듈을 소개하고 Qlib가 모듈을 수용하는 방법을 보여주어 기존 양적 연구의 몇 가지 관련 예를 제공합니다.

원시 데이터를 쿼리하고 처리하는 데이터 엔진을 제공하는 왼쪽 하단의 데이터 서버 모듈로 시작합니다. 검색된 데이터를 사용하여 연구원은 데이터 향상 모듈에서 자신의 데이터 세트를 구축할 수 있습니다. 연구자들은 효과적인 요소/기능을 탐색하고 구성하여 더 나은 데이터 세트를 구축하기 위해 많은 솔루션을 시도했습니다[Potvin et al., 2004; 널리 등, 1997; 알렌과 카잘라이안, 1999; Kakushadze, 2016]. 교육용 데이터 세트 생성[Feng et al., 2019]은 데이터세트 솔루션을 제공하기 위한 또 다른 연구 방향입니다. 모델 생성기 모듈은 데이터 세트를 기반으로 모델을 학습합니다. 최근 몇 년 동안 수많은 연구자들이 금융 데이터 세트에서 거래 신호를 마이닝하기 위해 모든 종류의 모델을 탐색했습니다[Sezer et al., 2019]. 또한, 학습을 시도하는 메타 학습[Vilalta and Drissi, 2002]은 Model Creator 모듈에 대한 새로운 학습 패러다임을 제공합니다. 현대 연구 워크플로우에서 재무 데이터를 모델링하는 방법이 많기 때문에 모델 관리 시스템은 필수가 되었습니다.

¹ https://en.wikipedia.org/wiki/플랫폼_파일_데이터베이스

워크플로의 필수 부분입니다. 모델 관리자 모듈은 현대의 정량적 연구자를 위해 이러한 문제를 처리하도록 설계되었습니다. 앙상블 학습은 모델이 다양하여 기계 학습 모델의 성능과 견고성을 향상시키는 데 상당히 효과적인 방법이며 금융 분야에서 자주 사용됩니다[Qiu et al., 2014; 양 외, 2017; Zhao 등, 2017]. Model Ensemble 모듈에서 지원합니다. Portfolio Generator 모듈은 포트폴리오 관리[Qian et al., 2007]로 알려진 모델에서 출력되는 거래 신호로부터 포트폴리오를 생성하는 것을 목표로 합니다. Barra[Sheikh, 1996]는 이 작업에 가장 많이 사용되는 솔루션을 제공합니다. 대상 포트폴리오와 함께 전략의 성과를 검사하기 위한 고충실도 거래 시뮬레이터인 Orders Executor 모듈과 거래 신호, 포트폴리오 및 실행 결과를 자동으로 분석하는 Analyzer 모듈을 제공합니다. Order Executor 모듈은 분석기 모듈에서 생성된 환경의 피드백이 필요한 일부 학습 패러다임(예: RL)에 대한 인프라 구조를 제공할 수 있는 백 테스트 기능이 아닌 반응형 시뮬레이터로 설계되었습니다.

양적 투자의 데이터는 매트에 대한 시계열이며 시간별로 업데이트됩니다. 샘플 내 데이터 세트의 크기는 시간에 따라 증가합니다. 새 데이터를 활용하는 일반적인 방법은 모델을 정기적으로 업데이트하는 것입니다[Wang et al., 2019b]. 증가하는 샘플 내 데이터의 더 나은 활용 외에도 동적으로 업데이트되는 모델[Yang et al., 2019] 및 거래 전략[Wang et al., 2019a]은 주식 시장의 동적 특성으로 인해 성능을 더욱 향상시킬 것입니다[Adam et al., 2016].

따라서 Static Workflow에서 일련의 정적 모델 및 거래 전략을 사용하는 것은 분명히 최적의 솔루션이 아닙니다. 모델 및 전략의 동적 업데이트는 양적 투자에서 중요한 연구 방향입니다. 동적 모델링의 모듈은 이러한 솔루션을 수용할 수 있는 인터페이스와 인프라를 제공합니다.

3.3 고성능 인프라

재무 데이터

이 섹션에서는 양적 연구의 데이터 요구 사항을 요약합니다. 양적 연구에서 가장 자주 사용되는 데이터 형식은 이러한 형식을 따릅니다.

$$\text{BasicDataT} = \{x_i, t, a\}, i \in \text{Inst}, t \in \text{Time}, a \in \text{Attr}$$

여기서 x_i, t, a 는 기본 유형의 값(예: float, int), Inst는 금융 상품 집합(예: 주식, 옵션 등), Time은 타임 스탬프 집합(예: 주식 시장의 거래일)을 나타냅니다. Attr은 상품의 가능한 속성 세트(예: 시가, 거래량, 시장 가치)를 나타내고, T는 데이터의 최신 타임스탬프(예: 최신 거래 날짜)를 나타냅니다. x_i, t, a 는 시간 t에서 상품 i의 속성 a 값을 나타냅니다.

게다가 상품 풀은 시간이 지남에 따라 변화하는 일련의 금융 상품을 저장하는데 필요한 정보입니다.

$$\text{PoolT} = \{\text{poolt}\}, t \in \text{시간}, \text{poolt} \subseteq \text{Inst}$$

S&P 500 지수⁴ 풀의 전형적인 예입니다.

데이터 업데이트는 필수 기능입니다. 기존 기록 데이터는 시간이 지나도 변경되지 않습니다. 추가 작업만

⁴https://en.wikipedia.org/wiki/S%26P_500_인덱스

새로운 데이터가 필요합니다. 공식화된 업데이트 작업은 다음과 같습니다.

$$\begin{aligned} \text{BasicDataT} &= \text{OldBasicDataT} \cup \{x_i, t, a\} \\ \text{기본데이터T} + 1 &= \text{기본데이터T} \cup \{x_i, T + 1, a\} \\ \text{풀} + 1 &= \text{풀T} \cup \{\text{풀} + 1\} \end{aligned}$$

사용자 쿼리는 다음과 같이 형식화할 수 있습니다.

$$\begin{aligned} \text{DataQuery} &= \{x_i, t, a\} \in \text{poolt}, \text{poolt} \in \text{PoolQuery } a \in \\ &\text{Attrquery}, \text{timestart} \leq t \leq \text{timeend} \end{aligned}$$

특정 풀의 특정 시간 범위에서 약기의 일부 속성에 대한 데이터 쿼리를 나타냅니다.

이러한 요구 사항은 매우 간단합니다. 많은 상용 오픈 소스 솔루션이 이러한 작업을 지원합니다. 세 가지 범주로 분류하고 각 범주에서 널리 사용되는 구현을 나열합니다. • 범용 데이터베이스: MySQL[MySQL, 2001],

몽고디비[초도로우, 2013]

• 시계열 데이터베이스: InfluxDB [Naqvi et al., 2017] • 과학 컴퓨팅을

위한 데이터 파일: numpy[Oliphant, 2006] 배열 또는 pandas[McKinney, 2011] 데이터 프레임으로 구성된 데이터

범용 데이터베이스는 다양한 형식과 구조의 데이터를 지원합니다. 게다가 인덱싱, 트랜잭션, 엔터티 관계 모델 등과 같은 많은 정교한 메커니즘을 제공합니다. 대부분은 특정 시나리오에서 핵심 문제를 해결하기보다 특정 작업에 과도한 종속성과 불필요한 복잡성을 추가합니다. 시계열 데이터베이스는 시계열 데이터에 대한 데이터 구조 및 쿼리를 최적화합니다. 그러나 그들은 여전히 정량적 연구를 위해 설계되지 않았습니다. 여기서 데이터는 일반적으로 하드웨어 가속을 활용하기 위한 과학적 계산을 위한 압축 배열 기반 형식입니다. 데이터가 형식 변환 없이 디스크에서 클라이언트 끝까지 컴팩트한 어레이 기반 형식을 유지한다면 많은 시간을 절약할 수 있습니다. 그러나 범용 데이터베이스와 시계열 데이터베이스 모두 범용성을 위해 다른 형식으로 데이터를 저장하고 전송하므로 과학적 계산에 비효율적입니다.

데이터베이스의 비효율성으로 인해 어레이 기반 데이터는 과학계에서 인기를 얻고 있습니다. Numpy 배열 및 pandas 데이터 프레임은 디스크에 HDF5 또는 pickle6으로 저장되는 경우가 많은 과학적 계산의 주류 구현입니다. 이러한 형식의 데이터는 종속성이 적고 과학적 컴퓨팅에 매우 효율적입니다. 그러나 이러한 데이터는 단일 파일에 저장되며 업데이트하거나 쿼리하기 어렵습니다.

위의 스토리지 솔루션을 조사한 결과 정량적 연구 시나리오에 잘 맞는 솔루션은 없었습니다. 정량적 연구를 위한 맞춤형 솔루션 설계가 필요합니다.

파일 스토리지 설계 그림 2

는 파일 스토리지 설계를 보여줍니다. 그림의 왼쪽 부분에 표시된 것처럼 Qlib는 파일을 트리 구조로 구성합니다. 데이터는 다른 기준에 따라 폴더와 파일로 구분됩니다.

⁵https://en.wikipedia.org/wiki/계측_데이터_형식

⁶<https://docs.python.org/3/library/pickle.html>

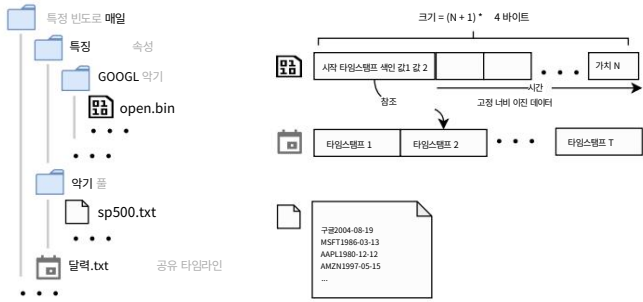


그림 2: 플랫폼 파일 데이터베이스에 대한 설명입니다. 왼쪽 부분은 파일 구조입니다. 오른쪽 부분은 파일의 내용입니다

주파수, 악기 및 속성. attribus의 모든 값은 바이트 단위로 인덱싱이 가능하도록 mat에 대해 컴팩트한 고정 너비의 바이너리 데이터로 저장됩니다. 공유 타임라인은 "calendar.txt"라는 파일에 별도로 저장됩니다.

속성 값의 데이터 파일은 처음 4바이트를 타임라인의 인덱스 값으로 설정하여 일련의 데이터의 시작 타임스탬프를 나타냅니다. 시작 시간 인덱스를 사용하여 Qlib는 시간 차원의 모든 값을 정렬할 수 있습니다.

데이터는 간결한 형식으로 저장되어 과학적 계산을 위해 배열로 결합하기에 효율적입니다. 과학적 계산에서 배열 기반 데이터와 같은 고성능을 달성하는 동시에 양적 투자 시나리오에서 데이터 업데이트 요구 사항을 충족합니다. 모든 데이터는 시간 순으로 정렬됩니다. 새로운 데이터는 appending에 의해 업데이트될 수 있으며 이는 매우 효율적입니다. 속성 또는 악기를 추가하고 제거하는 것은 별도의 파일에 저장되기 때문에 매우 간단하고 효율적입니다. 이러한 디자인은 매우 가법입니다. 데이터베이스 오버헤드 없이 Qlib는 높은 성능을 달성합니다.

Expression Engine 기

본 데이터를 기반으로 새로운 요소/특징을 개발하는 것은 상당히 일반적인 작업입니다. 이러한 작업은 많은 정량적 연구자들에게 상당한 시간이 소요됩니다. 둘 다 이러한 요소를 코드로 구현하고 계산 프로세스는 시간이 많이 걸립니다. 따라서 Qlib는 이러한 작업의 노력을 최소화하기 위해 표현식 엔진을 제공합니다.

사실 요인/특징의 성질은 기본 데이터를 목표값으로 변환하는 기능이다. 함수는 일련의 표현식 조합으로 분해될 수 있습니다. 익스프레션 엔진은 이 아이디어를 기반으로 설계되었습니다. 이 표현 엔진을 통해 양적 연구자는 복잡한 코드 대신 표현식을 작성하여 새로운 요소/기능을 구현할 수 있습니다. 예를 들어 Bollinger Band 기술 지표 [Bollinger, 2002]는 널리 사용되는 기술 요소이며 그 상한은 간단한 표현으로 구현할 수 있습니다.”(MEAN(\$close, N) + 2*STD(\$close, N) - \$close)/MEAN(\$close, N)”을 식 엔진과 함께 사용합니다.

이러한 구현은 간단하고 읽기 쉽고 재사용 및 유지 관리가 가능합니다. 일련의 간단한 표현식만으로 데이터 세트를 쉽게 구축할 수 있습니다. 효과적인 거래 신호를 구성하기 위한 검색 표현은 많은 연구자들에 의해 탐구된 전형적인 연구 주제입니다[Allen and Karjalainen, 1999; 널리 등, 1997; 포트빈 등, 2004]. 표현

엔진은 이러한 연구 주제에 필수적인 도구입니다.

캐시 시스템

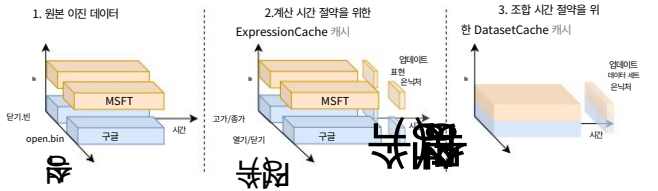


그림 3: Qlib의 디스크 캐시 시스템; 표현식 계산 시간을 절약하기 위한 표현식 캐시; 데이터 조합 시간을 절약하기 위한 데이터셋 캐시

복제된 계산을 피하기 위해 Qlib에는 내장 캐시가 있습니다. 체계. 메모리 캐시와 디스크 캐시로 구성됩니다.

메모리 내 캐시 Qlib가 표현식 엔진으로 요소/기능을 계산할 때 표현식을 구문 분석 트리로 구문 분석합니다. 노드의 모든 계산 결과는 메모리의 LRU(최소 사용) 캐시에 저장됩니다. 동일한 (하위)식의 복제 계산을 저장할 수 있습니다.

디스크 캐시 양적 투자에서 데이터 처리의 일반적인 워크플로우는 세 단계로 나눌 수 있습니다. 원본 데이터 가져오기, 표현 계산 및 과학적 계산을 위해 데이터를 어레이로 결합. 식을 계산하고 데이터를 결합하는 데는 시간이 많이 걸립니다. 공유 중간 데이터를 캐시할 수 있다면 많은 시간을 절약할 수 있습니다. 실제 데이터 처리 작업에서 많은 중간 결과를 공유할 수 있습니다. 예를 들어, 동일한 표현식 계산을 다른 데이터 처리 작업에서 공유할 수 있습니다. 따라서 Qlib는 2레벨 디스크 캐시 메커니즘을 설계했습니다. 캐시 시스템은 그림 3에 나와 있습니다. 왼쪽 부분은 섹션 3.3에서 설명한 원본 데이터입니다. 첫 번째 수준은 계산된 모든 표현식을 디스크 캐시에 저장하는 표현식 캐시입니다. 식 캐시의 데이터 구조는 원본 데이터와 동일합니다. 표현식 캐시를 사용하면 동일한 표현식이 한 번만 계산됩니다. 식 캐시 다음에는 조합 시간을 절약하기 위해 결합된 데이터를 저장하는 데이터 세트 캐시가 있습니다. 두 레벨의 캐시 데이터는 시간별로 정렬되고 시간 차원에서 인덱싱 가능하므로 쿼리 시간이 변경되더라도 디스크 캐시를 공유할 수 있습니다. 또한 Qlib는 시간별 데이터 정렬 덕분에 새로운 데이터를 추가하여 데이터 업데이트를 지원합니다. 이러한 메커니즘을 사용하면 데이터의 유지 관리가 훨씬 쉬워집니다.

3.4 기계 학습에 대한 지침 섹션 2에서 논의한 것처럼 기

계 학습 알고리즘에 대한 지침은 매우 중요합니다. Qlib는 기계 학습 알고리즘을 위한 일반적인 데이터 세트를 제공합니다. 데이터 전처리, 학습 대상 등과 같은 몇 가지 일반적인 작업 설정은 Qlib에서 찾을 수 있습니다. 연구원은 처음부터 모든 것을 탐색할 필요가 없습니다. 대신, Qlib는 연구자들이 다양한 데이터셋을 제공합니다.

대부분의 기계 학습 알고리즘에서 하이퍼파라미터 최적화는 더 나은 일반화를 달성하는 데 필요한 단계입니다. 중요하지만 많은 노력이 필요하고

	HDF5	MySQL	MongoDB	InfluxDB	Qlib -E -D	Qlib +E -D	Qlib +E +D	
스토리지(MB)	287	1,332	911	394	303	802	1,000	
데이터 로드	0.80 ± 0.22	182.5 ± 4.2	70.3 ± 4.9	186.5 ± 1.5	0.95 ± 0.05	4.9 ± 0.07		7.4 ± 0.3
만료 계산		179.8 ± 4.4			137.7 ± 7.6	35.3 ± 2.3		-
인덱스 변환		-			3.6 ± 0.1			-
폴로 필터링			3.39 ± 0.24					-
데이터 결합			1.19 ± 0.30					-
합계(1CPU)(초)	184.4 ± 3.7	365.3 ± 7.5	253.6 ± 6.7	368.2 ± 3.6	147.0 ± 8.8	47.6 ± 1.0		7.4 ± 0.3
총(64CPU) (s)		-			8.8 ± 0.6	4.2 ± 0.2		-

표 1: 다양한 스토리지 솔루션의 성능 비교

꽤 반복적입니다. 따라서 Qlib는 이러한 작업을 보다 쉽게 수행할 수 있도록 HTE(Hyperparameter Tuning Engine)를 제공합니다. HTE는 하이퍼파라미터 검색 공간 Θ 를 정의한 다음 최적의 하이퍼파라미터 θ 를 자동으로 검색하는 인터페이스를 제공합니다.

시계열 데이터를 모델링하는 일반적인 재무 작업에서 새 데이터는 시간별로 순차적으로 제공됩니다. 새로운 데이터를 활용하려면 모델을 주기적으로 새로운 데이터로 재교육해야 합니다.

새로운 최상의 하이퍼파라미터 θ 는 변경되지만 종종 이전의 최상의 하이퍼파라미터에 가깝습니다. HTE는 재무 작업에 대한 하이퍼파라미터 최적화 전용 메커니즘을 제공합니다.

더 적은 시도로 최상의 지점에 도달할 수 있는 더 나은 기회를 위해 하이퍼파라미터 검색 공간에 대한 새로운 분포를 생성합니다. 검색 θ 분포는 다음과 같이 공식화될 수 있습니다.

$$p_{\text{new}}(\mathbf{x}) = \text{Exp} \left(\frac{p_{\text{prior}}(\mathbf{x}) \cdot \theta_{\text{prev}, \sigma^2}(\mathbf{x})}{p_{\text{prior}}(\mathbf{x}) + \theta_{\text{prev}, \sigma^2}(\mathbf{x})} \right)$$
여기서 p_{prior} 는 원래 하이퍼파라

미터 검색 공간입니다. $\theta_{\text{prev}, \sigma^2}(\mathbf{x}) \sim N(\theta_{\text{prev}}, \sigma^2)$; θ_{prev} 는 마지막 모델 학습에서 최고의 하이퍼파라미터입니다. 하이퍼파라미터 검색 공간의 영역은 동일하게 유지되지만 θ_{prev} 주변의 확률 밀도는 증가합니다.

4 사용 사례 및 성능 평가

4.1 사용 사례 Qlib는

구성 기반 파이프라인 엔진(CDPE)을 제공하여 연구자가 그림 1에 표시된 전체 연구 워크플로를 보다 쉽게 구축할 수 있도록 도와줍니다. 사용자는 List ??와 같은 간단한 구성 파일만으로 워크플로를 정의할 수 있습니다 (일부 사소한 세부 사항은 "... "로 대체됨). 이러한 인터페이스는 필수 사항이 아니며 빌딩 블록과 같은 코드로 정량적 연구 워크플로를 구축할 수 있도록 사용자에게 최대한의 유연성을 제공합니다.

4.2 성능 평가

데이터 처리 성능은 AI 기술과 같은 데이터 기반 방법에 중요합니다. AI 지향 플랫폼인 Qlib는 데이터 저장 및 데이터 처리를 위한 솔루션을 제공합니다. Qlib의 성능을 시연하기 위해 HDF5, MySQL, MongoDB, InfluxDb 및 Qlib를 포함하여 섹션 3.3에서 설명한 여러 다른 솔루션과 Qlib를 비교합니다. Qlib +E -D는 표현식 캐시가 활성화되고 데이터 세트 캐시가 비활성화된 Qlib를 나타냅니다.

```
model:
  class: "qlib.model.GBDTModel"
  args: ...
data:
  class: "qlib.dataset.Alpha360"
learner:
  class: "qlib.trainer.NormalTrainer"
  args: ...
portfolio_manager:
  class: "qlib.portfolio.TopkStrategy"
executor:
  account: ...
```

그림 4: CDPE의 구성 예

솔루션의 작업은 데이터 쿼리 및 처리를 포함하는 주식 시장의 기본 OHLCV7 일일 데이터에서 데이터 세트를 만드는 것입니다. 최종 데이터 세트는 OHLCV 데이터 (예: "Std(\$close, 5)/\$close")에서 파생된 14개의 요인/특성으로 구성됩니다. 데이터의 시간 범위는 2007년 1월 1일부터 2020년 1월 1일까지입니다. 주식 풀은 매일 800개의 주식으로 구성되며 매일 변경됩니다.

- 각 솔루션의 총 시간을 비교하는 것 외에도 자세한 내용을 위해 작업을 다음 단계로 나눕니다.
- 데이터 로드 과학적 계산을 위한 배열 기반 형식으로 OHCLV 데이터 또는 캐시를 RAM에 로드합니다.
 - 계산 Expr. 파생 요인/특징을 계산합니다.
 - 색인 변환 Qlib에만 적용됩니다. Qlib는 원래 데이터에 인덱스(즉, 타임스탬프, 주식 ID)를 저장하지 않기 때문에 데이터 인덱스를 설정해야 합니다.
 - 데이터 필터링 특정 풀로 주식 데이터를 필터링합니다. 예를 들어 SP500은 총 1,000개 이상의 주식을 포함하지만 매일 500개의 주식만 포함합니다. 특정 날짜에 SP500에 포함되지 않은 데이터는 SP500에 포함된 적이 있지만 필터링해야 합니다. 일부 파생 기능은 과거 OHLCV 데이터에 의존하기 때문에 데이터를 로드할 때 데이터를 필터링하는 것은 불가능합니다.

- 데이터 결합 서로 다른 모든 데이터를 연결합니다. 배열 기반 데이터의 단일 조각으로 저장

표 1에서 볼 수 있듯이 Qlib의 콤팩트 스토리지는 과학 전용 스토리지와 비슷한 크기와 로딩 속도를 달성합니다.

7주식의 시가, 고가, 저가, 종가 및 거래량

HDF5 데이터 파일, 데이터베이스가 데이터를 로드하는 데 너무 많은 시간이 걸립니다. 기본 구현을 살펴본 후 데이터가 범용 데이터베이스와 시계열 데이터베이스 솔루션 모두에서 너무 많은 인터페이스 계층과 불필요한 형식 변환을 거침을 발견했습니다. 이러한 오버헤드는 데이터 로드 프로세스를 크게 저하시킵니다. Qlib의 메모리 캐시로 인해 Qlib -E -D는 Compute Expr 시간의 약 24%를 절약합니다. 또한 Qlib는 식 캐시 및 데이터 세트 캐시 메커니즘을 제공합니다. Qlib +E -D에서 표현식 캐시를 활성화하면 Compute Expr 시간의 80.4%가 됩니다. 누락된 표현식 캐시가 없으면 저장됩니다. 요소/기능을 각 주식에 대한 하나의 배열 기반 데이터로 결합하면 Compute Expr에 포함된 Qlib +E -D의 주요 시간 소모를 설명합니다. 단계, 계산 비용 외에 가장 시간이 많이 걸리는 단계는 데이터 조합입니다.

데이터 세트 캐시는 이러한 오버헤드를 줄이기 위해 설계되었습니다. Qlib + E + D 열에 표시된 것처럼 시간 비용이 더 줄어듭니다.

또한 Qlib는 여러 CPU 코어를 활용하여 계산을 가속화할 수 있습니다. 표 1의 마지막 줄에서 볼 수 있듯이 CPU가 여러 개인 Qlib의 경우 시간 비용이 크게 줄어듭니다. Qlib + E + D는 기존 캐시를 읽기만 하고 거의 아무것도 계산하지 않기 때문에 더 이상 가속할 수 없습니다.

4.3 Qlib에 대한 추가 정보 Qlib는 지

속적으로 개발 중인 오픈 소스 플랫폼입니다.

더 자세한 문서는 [github 저장소](#)에서 찾을 수 있습니다. 이 백서에서 자세히 소개하지 않은 많은 기능(예: 클라이언트용 라이브러리, 저장소에 대한 접근 권한, 복제 및 브랜칭, 클라이언트 통합)이 있습니다.

5. 결론

본 논문에서는 인공지능 시대 현대 정량연구자들의 실질적인 문제를 제시한다. 이러한 실용적인 문제를 기반으로 우리는 모든 퀀트 연구원이 퀀트 투자에서 AI 기술의 큰 잠재력을 실현할 수 있도록 하는 것을 목표로 하는 Qlib을 설계하고 구현합니다.

참조

[Adam et al., 2016] Klaus Adam, Albert Marcet, Juan Pablo Nicolini. 주식 시장 변동성 및 학습, 2016.

[Allen 및 Karjalainen, 1999] Franklin Allen 및 Risto Karjalainen. 유전자 알고리즘을 사용하여 기술적 거래 규칙 찾기. 금융 경제학 저널, 51(2):245-271, 1999.

[볼린저, 2002] 존 볼린저. 볼린저 온 볼린저 밴드. 맥그로 힐 프로페셔널, 2002.

[초도로우, 2013] 크리스티나 초도로우. MongoDB: 최종 가이드: 강력하고 확장 가능한,, 데이터 스토리지.
오라일리 미디어, Inc.”, 2013.

[Deng et al., 2016] Yue Deng, Feng Bao, Youyong Kong, Zhiquan Ren, Qionghai Dai. 재무 신호 표현 및 거래를 위한 심층 직접 강화 학습. 신경망 및 학습 시스템에 대한 IEEE 트랜잭션, 28(3):653–664, 2016.

[Feng et al., 2019] Fuli Feng, Huimin Chen, Xiangnan He, Ji Ding, Maosong Sun, Tat-Seng Chua. 적대적 훈련으로 주식 이동 예측을 향상시킵니다. 제28차 인공지능에 관한 국제 합동 회의 절차, 5843–5849페이지. AAAI 보도, 2019.

[퍼스, 2004] 엔퍼스. Quantlib를 사용하는 이유 좋아: <http://www.quantlib.org>. 공동. 영국/출판물/quantlib. PDF, 2004.

[카쿠샤제, 2016] 주라 카쿠샤제. 101 공식 알파. Wilmott, 2016(84):72-81, 2016.

[Li et al., 2016] Bin Li, Doyen Sahoo, Steven CH Hoi.
Olps: 온라인 포트폴리오 선택을 위한 도구 상자. 기계 학습 연구 저널, 17(1):1242–1246, 2016.

[맥키니, 2011] 웨스 맥키니. pandas: 데이터 분석 및 통계를 위한 기본 Python 라이브러리입니다. 고성능 및 과학적 컴퓨팅을 위한 Python, 14, 2011.

[머피, 1999] 존 J 머피. 금융 시장의 기술적 분석: 거래 방법 및 응용 프로그램에 대한 포괄적인 가이드. 핑귌, 1999.

[MySQL, 2001] AB MySQL. 마이SQL, 2001.

[Naqvi et al., 2017] Syeda Noor Zehra Naqvi, Sofia Yfanti dou 및 Esteban Zimanyi. 시계열 데이터베이스 및 in-fluxdb. Studienarbeit, Universite Libre de Bruxelles 2017.

[Neely et al., 1997] Christopher Neely, Paul Weller 및 Rob Dittmar. 외환 시장의 기술적 분석은 수익성이 있습니까? 유전자 프로그래밍 접근법. 재무 및 양적 분석 저널, 32(4):405-426, 1997.

[올리펀트, 2006] 트레이비스 E 올리펀트. NumPy 가이드, volume 1. Trelgol Publishing USA, 2006.

[페트코바, 2006] 칼리차 페트코바. Fama-French 요인이 예측 변수의 혁신을 대신합니까? 금융 저널, 61(2):581-612, 2006.

[Potvin et al., 2004] Jean-Yves Potvin, Patrick Soriano, Maxime Vallee. 유전자 프로그래밍으로 주식 시장에서 거래 규칙 생성. *Computers & Operations Research*, 31(7):1033-1047, 2004.

[Qian et al., 2007] Edward E Qian, Ronald H Hua, Eric H Sorensen. 정량적 자산 포트폴리오 관리: 최신 기술 및 응용 프로그램. CRC 출판부, 2007.

[Qiu et al., 2014] Xueheng Qiu, Le Zhang, Ye Ren, Pon nuthurai N Suganthan, Gehan Amaratunga. 화귀 및 시계열 예측을 위한 앙상블 딥 러닝.

양상을 학습(CIEL)의 계산 지능에 관한 2014 IEEE 심포지엄, 1-6페이지. IEEE, 2014.

⁸<https://github.com/microsoft/qlib/>

[Sezer et al., 2019] Omer Berat Sezer, Mehmet Ugur Gudelek 및 Ahmet Murat Ozbayoglu. 딥 러닝을 사용한 재무 시계열 예측: 체계적 문헌 검토: 2005-2019. arXiv 프리프린트 arXiv:1911.13288, 2019.

[세이크, 1996] 아미르 세이크. Barra의 위험 모델. Barra Research Insights, 1-24페이지, 1996.

[Vilalta와 Drissi, 2002] Ricardo Vilalta와 Youssef Drissi. 메타 학습에 대한 관점과 조사. 인공 지능 검토, 18(2):77-95, 2002.

[Wang et al., 2019a] Lewen Wang, Weiqing Liu, Xiao Yang, Jiang Bian. 보수적입니까 아니면 공격적입니까? 자신감 인식 동적 포트폴리오 구성. 신호 및 정보 처리에 관한 2019 IEEE 글로벌 컨퍼런스(GlobalSIP), 1-5페이지. IEEE, 2019.

[Wang et al., 2019b] Shouxiang Wang, Xuan Wang, Shaomin Wang, Dan Wang. 단기 부하 예측을 위한 어텐션 메커니즘 및 롤링 업데이트를 기반으로 한 양방향 장단기 메모리 방법. 국제 전력 및 에너지 시스템 저널, 109:470-479, 2019.

[Yang et al., 2017] Bing Yang, Zi-Jia Gong, and Wenqi Yang. 심층 신경망 앙상블을 이용한 주식 시장 지수 예측. 2017년 제36회 중국 통제 회의(CCC), 3882-3887페이지. IEEE, 2017.

[Yang et al., 2019] Xiao Yang, Weiqing Liu, Lewen Wang, Cheng Qu 및 Jiang Bian. 여러 투자 전략의 관심 기반 조합을 위한 분할 정복 프레임 워크. 신호 및 정보 처리(GlobalSIP)에 관한 2019 IEEE 글로벌 컨퍼런스, 1-5페이지.

IEEE, 2019.

[Zhao et al., 2017] Yang Zhao, Jianping Li, Lean Yu. 원유 가격 예측을 위한 딥러닝 앙상블 접근법. 에너지 경제학, 66:9-16, 2017.